

**REMARKS**

Claims 1-14 stand rejected under 35 U.S.C. § 103(a) as being obvious over U.S. Patent 6,434,649 to Baker et al. (hereinafter, Baker), in view of U.S. Patent Application Publication 2003/0126056 to Hausman et al. (hereinafter, Hausman) and further in view of U.S. Patent No. 5,983,270 to Abraham et al. (hereinafter, Abraham).

Applicants respectfully disagree with these rejections, and reconsideration is respectfully requested.

The Office Action substantially maintains the rejections under § 103 that were previously asserted in the Office Action mailed on January 6, 2009. Notwithstanding the response to Applicants' arguments provided in the present Office Action, Applicants maintain that no combination of the cited references Baker, Hausman and Abraham teaches or suggests the invention as recited in Claims 1-14 as previously presented. The arguments presented in the Office Action are addressed below.

The combination of Baker, Hausman and Abraham is not configured to recognize the record and field structure of non-field delineated data.

Claim 1 recites a "Programmable Streaming Data Processor" (PSDP) that includes a "data engine" configured to "recognize the record and field structure of non-field delineated data" and "determine field boundaries in the non-field delineated data." As described in paragraph [0015] of Applicants' published application:

...the PSDP can parse non-field-delineated, streaming data from the [streaming data source] into block header fields, record header fields, and record data fields... In other words, the PSDP can be programmed to understand the record and field structure of the data which the analysis software running on the CPU of the JPU wishes to analyze. Therefore, the PSDP can further process data in the specific format of the database application. (Emphasis added.)

The Office Action asserts that such a "data engine" is taught by Hausman, referring to the application programming interface (API) 104 in Fig. 1a in Hausman. However, Hausman's API 104 cannot recognize or determine field boundaries in non-field-delineated data because it does not receive non-field-delineated data. On the contrary, the data received by Hausman's API 104

is already field-delineated. As described in paragraph [0045] of Hausman with relation to data from source 101 received through queue 108:

API 104 reviews data records included within the stream [received from source 101 through queue 108]...and selects...data records requested by users served by the API. (Emphasis added.)

This interpretation of Hausman is actually supported by the Examiner's admission at the bottom of page 2 of the Advisory Action dated July 28, 2008 that "the output data [as received by API 104 from queue 108] includes delimiters between records and between elements within the records." (Emphasis added.) Thus, the data received by Hausman's API 104 is field-delineated, and therefore the API 104 does not receive non-field delineated data.

In the Office Action "Response to Arguments" on page 9, section 9, the Examiner replied to the argument above:

It is noted that paragraph [0039] of Hausman states "Data is preferably received live from system applications 122 by queue 123; delimiters are inserted, if necessary, between records and optionally between individual elements within records." Thus, in order for the delimiters to be inserted, the record and structure of the stream has to be recognized.

However, paragraph [0039] of Hausman is not referring to the API 104. Rather, it is referring to the system server 120, which is located at the data stream source 101 across the network. As shown in Fig. 1a of Hausman, the data stream source 101 (at left) includes a system server 122 that retrieves data from a database 121 and broadcasts the data across the network to a client/user system 102 (at right), where it is received by a client server 105 (para. [0040]-[0043]). As stated in paragraph [0039], the system server 120 provides field-delineated data to the client server 105 (at which the API 104 resides):

Server system 120 comprises one or more source application systems 122 ... and queue 123. Data created and processed by application(s) 122, and optionally stored within database(s) 121, is forwarded to queue 123 and entered into a preferably continuous, semi-continuous, or intermittent live data stream... Data is preferably received live from system applications 122 by queue 123; delimiters are inserted, if necessary, between records and optionally between individual elements within records, as for example according to the... FINANCIAL INFORMATION EXCHANGE PROTOCOL (FIX), Version 4.2 ...and data is sent to or made available for reading by client systems. As soon as data has been formatted in the proper format within the queue [123], records,

separated by delimiters, are sequentially broadcast ... to client systems [105] that have been enabled and/or authorized to receive them.

The document cited by Hausman, FIX Version 4.2 (May 1, 2001), describes the field-delineated data used by Hausman in more detail:

The general format of a FIX message is a standard header followed by the message body fields and terminated with a standard trailer. Each message is constructed of a stream of <tag>=<value> fields with a field delimiter between fields in the stream.

(FIX 4.2 with Errata 20010501, page 9.)

All fields (including those of data type data i.e. SecureData, RawData, SignatureData, XmlData, etc.) in a FIX message are terminated by a delimiter character. The non-printing, ASCII "SOH" (#001, hex: 0x01, referred to in this document as <SOH>), is used for field termination.

(FIX 4.2 with Errata 20010501, page 10.)

Hausman's field-delineated data is shown in further detail in Fig. 3. Here, the data records 301 (at top) each contain a number of data items 302, and are received by the source queue 123 (para. [0060]-[0061]). The source queue 123 inserts delimiters 304 between the data items 303 to generate a field-delineated data stream (at center) to the API 104. Thus, the queue 123 at the system server 120 provides field-delineated data to the API 104.

Accordingly, Hausman's API 104 receives field-delineated data, and so is not arranged to "recognize the record and field structure of the non-field delineated data" nor "determine field boundaries in the non-field delineated data" as recited in Claim 1. Because Hausman's data stream (between the source queue 123 and the API 104) is field-delineated, he also fails to teach "a streaming data interface, arranged to receive non-field delineated data from a streaming data source, and "a streaming interface First In First Out (FIFO), arranged to temporarily store the streaming non-field-delineated data from the streaming data interface," as recited in Claim 1.

The combination of Baker, Hausman and Abraham neither selects fields to be assembled into output tuples nor assembles fields into tuples

The combination of Baker, Hausman and Abraham neither selects fields to be assembled into output tuples nor assembles fields into tuples as required by Applicants' Claim 1. As described in paragraph [0108] of Applicants' published application, the term "tuple" is used by

Applicants for the purpose of differentiating “raw” disk 23 record formats from PSDP 28 output record formats. For example, paragraph [0019] of Applicants’ published application teaches that an output tuple “is comprised of the fields of the source record from the disk that are to be selected for further processing by the CPU and PSDP generated fields.”

The Examiner has likened Hausman’s API 104 to Applicants’ claimed data engine. However, as described in paragraph [0045] of Hausman, the API 104 “selects...data records requested by users served by the API” (emphasis added) and passes complete records to clients requesting them. Thus, Hausman describes wholesale selection by the API 104 of entire requested records (i.e., raw data), and not selection of particular fields of those records as recognized in the non-field delineated data by the data engine (i.e., an output tuple comprised of the fields of the source record from the disk that are to be selected for further processing by the CPU and PSDP generated fields). Moreover, by the Examiner’s own admission, “according to [0045], only the records that are requested by the users are sent to that particular user [JPU] for processing.” (Emphasis added.) These records are the same as the records from the source. They are not comprised of fields selected for further processing. Hausman has no teaching, mention or even a suggestion of selecting fields to be assembled into output tuples as required by Applicants’ claims.

The above argument was presented in the previous Amendment filed April 6, 2009. In the Office Action “Response to Arguments” on page 9, section 10, the Examiner replied to the argument above:

Paragraph [0048] of Hausman states “...API 104 formats (or reformats), i.e., maps, records into any form(s) requested by the individual user(s) as for example by re-ordering, deleting, editing, and/or adding elements within the information strings carried by the records.” ...Therefore, the reformatting of the records assembles the received data into records with elements that meet the requirement of the user [process fields to select one or more fields to be assembled into output tuples].

However, the process of formatting at the API 104 does not constitute “process[ing] fields to select one or more fields to be assembled into output tuples” as recited in Claim 1. In particular, the formatting at the API 104 is not specific to particular fields; instead, the API merely formats complete records without selecting individual fields of the records for assembly. As stated in paragraph [0045] of Hausman, the API 104 “selects...data records requested by users served by

the API” (emphasis added) and passes complete records to clients requesting them. This process is shown in Fig. 3 of Hausman, where the API 104 outputs records to different users (“User 1,” “User 2,” “User 3”) in different formats according to each user’s preference (para. [0066]). Each record (at bottom) is therefore a reformatted version of an original data record 301 (at top) retrieved from the server database 121. For example, User 1 receives Data record 1 and Data record 2, which are reformatted versions of the original data records 301 (at top). Thus, Hausman describes wholesale selection by the API 104 of entire requested records (i.e., raw data), and does not select particular fields of those records because the record is selected in its entirety. In contrast, an output tuple is composed of fields to be selected for further processing, meaning that fields are selected based on the particular data included in each field. Hausman thus does not teach “select[ing] one or more fields to be assembled into output tuples” as recited in Claim 1.

Moreover, Hausman also does not assemble fields into an output tuple as recited in Applicants’ Claim 1. As described above, an output tuple “is comprised of the fields...that are to be selected for further processing by the CPU and PSDP generated fields” (paragraph [0019] of Applicants’ published application). These selected fields, plus those assembled by the claimed tuple generator, do not equate to a row of raw data from the data source as asserted by the Examiner. Rather, they comprise a new data form (*i.e.*, a tuple) generated by the tuple generator of the PSDP. A row of data, as in Hausman, equates to a database record but does not equate to a tuple. Therefore, Hausman also fails to teach assembling fields into output tuples.

In view of the above, it is respectfully submitted that the cited references cannot be combined to teach or suggest the invention as recited in Claim 1. The combination of Baker, Hausman and Abraham does not overcome any of the deficiencies of Hausman as described above.

Claims 2-14 depend from independent Claim 1 and thus the foregoing applies. As a result, the § 103 rejection of Claims 1-14 is believed to be overcome, and reconsideration is respectfully requested.

**Information Disclosure Statement**


An Information Disclosure Statement (IDS) is being filed concurrently herewith. Entry of the IDS is respectfully requested.

**CONCLUSION**

In view of the above amendments and remarks, it is believed that all claims are in condition for allowance, and it is respectfully requested that the application be passed to issue. If the Examiner feels that a telephone conference would expedite prosecution of this case, the Examiner is invited to call the undersigned.

Respectfully submitted,

HAMILTON, BROOK, SMITH & REYNOLDS, P.C.

By   
Benjamin J. Sparrow  
Registration No. 62,259  
Telephone: (978) 341-0036  
Facsimile: (978) 341-0136

Concord, MA 01742-9133

Date: 11/6/09